

EEEEEEEEE	RRRRRRRRR	FFFF
EEEEEEEEE	RRRRRRRRR	FFF
EEEEEEEEE	RRRRRRRRR	FFF
EEE	RRR	FFF
EEEEEEEEE	RRRRRRRRR	FFFF
EEEEEEEEE	RRRRRRRRR	FFFF
EEEEEEEEE	RRRRRRRRR	FFFF
EEE	RRR	FFF
EEEEEEEEE	RRR	FFF
EEEEEEEEE	RRR	FFF
EEEEEEEEE	RRR	FFF

0001 SUBROUTINE LPA11K (LUN)
0002
0003 C Version: 'V04-000'
0004 C*****
0005 C*
0006 C* (COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
0007 C* DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
0008 C* ALL RIGHTS RESERVED.
0009 C*
0010 C* THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
0011 C* ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
0012 C* INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
0013 C* COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
0014 C* OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
0015 C* TRANSFERRED.
0016 C*
0017 C* THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
0018 C* AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
0019 C* CORPORATION.
0020 C*
0021 C* DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
0022 C* SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
0023 C*
0024 C*
0025 C*****
0026 C
0027 C
0028 C
0029 C
0030 C AUTHOR BRIAN PORTER CREATION DATE 20-AUG-1979
0031 C
0032 C
0033 C++
0034 C Functional description:
0035 C
0036 C This module displays entries made for the lpa11k.
0037 C
0038 C Modified by:
0039 C
0040 C V03-002 SAR0077 Sharon A. Reynolds, 20-Jun-1983
0041 C Changed the carriage control in the 'format' statements
0042 C for use with ERF.
0043 C
0044 C V03-001 SAR0036 Sharon A. Reynolds, 8-Jun-1983
0045 C Removed brief/cryptic support.
0046 C
0047 C V02-004 BP0004 Brian Porter, 23-NOV-1981
0048 C Minor edit.
0049 C
0050 C V02-003 BP0003 Brian Porter, 06-NOV-1981
0051 C Added 'device attention' support.
0052 C
0053 C V02-002 BP0002 Brian Porter, 23-JUL-1981
0054 C Added different uba handling.
0055 C
0056 C V02-001 BP0001 Brian Porter, 29-JUN-1981
0057 C Removed call to UNUSED_BITS. Added call to LOGGER and

0058 C
0059 C**
0060 C--
0061
0062 INCLUDE 'SRC\$:MSGHDR.FOR /NOLIST'
0121 INCLUDE 'SRC\$:DEVERR.FOR /NOLIST'
0222
0223
0224 BYTE LUN
0225 BYTE MAINTENANCE(0:3)
0226
0227 INTEGER*4 FIELD
0228 INTEGER*4 COMPRESSC
0229 INTEGER*4 CNTRL_MAINT
0230 INTEGER*4 CNTRL_STATUS
0231 INTEGER*4 RDA
0232 INTEGER*4 UBAREG(2)
0233
0234 PARAMETER TIMEOUT = 96
0235
0236 EQUIVALENCE (CNTRL_MAINT,EMBSL_DV_REGSAV(0))
0237 EQUIVALENCE (CNTRL_STATUS,EMBSL_DV_REGSAV(1))
0238 EQUIVALENCE (RDA,EMBSL_DV_REGSAV(2))
0239 EQUIVALENCE (MAINTENANCE,EMBSL_DV_REGSAV(3))
0240 EQUIVALENCE (UBAREG,EMBSL_DV_REGSAV(4))
0241
0242 CHARACTER*7 V1CNTRL_MAINT(0:0)
0243 DATA V1CNTRL_MAINT(0)/'GO BIT'/
0244
0245
0246 CHARACTER*20 V2CNTRL_MAINT(6:11)
0247 DATA V2CNTRL_MAINT(6)/'IN INTERRUPT ENABLE'/
0248 DATA V2CNTRL_MAINT(7)/'IN READY'/
0249 DATA V2CNTRL_MAINT(8)/'STEP UP'/
0250 DATA V2CNTRL_MAINT(9)/'ROM 0'/
0251 DATA V2CNTRL_MAINT(10)/'ROM 1'/
0252 DATA V2CNTRL_MAINT(11)/'ENABLE ARBITRATION'/
0253
0254
0255 CHARACTER*11 V3CNTRL_MAINT(13:15)
0256 DATA V3CNTRL_MAINT(13)/'CRAM WRITE'/
0257 DATA V3CNTRL_MAINT(14)/'RESET'/
0258 DATA V3CNTRL_MAINT(15)/'RUN'/
0259
0260
0261 CHARACTER*21 V1CNTRL_STATUS(6:7)
0262 DATA V1CNTRL_STATUS(6)/'OUT INTERRUPT ENABLE'/
0263 DATA V1CNTRL_STATUS(7)/'OUT READY'/
0264
0265
0266 CHARACTER*31 FATAL_ERROR('340'0:'351'0)
0267 DATA FATAL_ERROR('340'0)/'SLAVE POWER LOW'/
0268 DATA FATAL_ERROR('341'0)/'MASTER FIFO R/W ERROR'/
0269 DATA FATAL_ERROR('342'0)/'I/O BUS SACK TIMEOUT'/
0270 DATA FATAL_ERROR('343'0)/'MASTER INITIAL CONDITION ERROR'/
0271 DATA FATAL_ERROR('344'0)/'MASTER/SLAVE VERSION ERROR'/
0272 DATA FATAL_ERROR('345'0)/'SLAVE COLD START TIMEOUT'/'

```

0273  DATA    FATAL_ERROR('346'0)/*FATAL SLAVE ERROR*/
0274  DATA    FATAL_ERROR('347'0)/*FAST PATH DATA ERRORS*/
0275  DATA    FATAL_ERROR('350'0)/*FIFO DATA ERROR*/
0276  DATA    FATAL_ERROR('351'0)/*ILLEGAL FATAL ERROR CODE*/
0277
0278
0279  CHARACTER*33  START_ERROR('300'0:'327'0)
0280  DATA    START_ERROR('300'0)/*NO ROOM FOR REQUEST*/
0281  DATA    START_ERROR('301'0)/*GO BIT SET WITH ''READY IN'' SET*/
0282  DATA    START_ERROR('302'0)/*MULTI-USER REQ IN DEDICATED MODE*/
0283  DATA    START_ERROR('303'0)/*ILLEGAL ''START ERROR'' CODE*/
0284  DATA    START_ERROR('304'0)/*DEDICATED REQ IN MULTI-USER MODE*/
0285  DATA    START_ERROR('305'0)/*ILLEGAL ''START ERROR'' CODE*/
0286  DATA    START_ERROR('306'0)/*START REQUEST WITHOUT INITIALIZE*/
0287  DATA    START_ERROR('307'0)/*ILLEGAL ''START ERROR'' CODE*/
0288  DATA    START_ERROR('310'0)/*MULTIPLE INITIALIZE COMMAND*/
0289  DATA    START_ERROR('311'0)/*ILLEGAL ''START ERROR'' CODE*/
0290  DATA    START_ERROR('312'0)/*STOP COMMAND AND USER NOT ACTIVE*/
0291  DATA    START_ERROR('313'0)/*ILLEGAL ''START ERROR'' CODE*/
0292  DATA    START_ERROR('314'0)/*ODD REQUEST DESCRIPTOR ARRAY ADDR*/
0293  DATA    START_ERROR('315'0)/*ILLEGAL ''START ERROR'' CODE*/
0294  DATA    START_ERROR('316'0)/*INITIALIZE WITH IMPROPER VERSION*/
0295  DATA    START_ERROR('317'0)/*ILLEGAL ''START ERROR'' CODE*/
0296  DATA    START_ERROR('320'0)/*DEVICE NOT PRESENT*/
0297  DATA    START_ERROR('321'0)/*ILLEGAL ''START ERROR'' CODE*/
0298  DATA    START_ERROR('322'0)/*ILLEGAL FUNCTION*/
0299  DATA    START_ERROR('323'0)/*ILLEGAL FUNCTION*/
0300  DATA    START_ERROR('324'0)/*REQUEST DESCRIPTOR ARRAY NXM*/
0301  DATA    START_ERROR('325'0)/*BUFFER/USER STATUS WORD ODD ADDR*/
0302  DATA    START_ERROR('326'0)/*DEVICE NOT PRESENT ON I/O BUS*/
0303  DATA    START_ERROR('327'0)/*XFER COUNT LESS THAN 256 WORDS*/
0304
0305  CHARACTER*33  USER_ERROR('240'0:'261'0)
0306  DATA    USER_ERROR('240'0)/*NON-FATAL ERROR COUNT EXCEEDED*/
0307  DATA    USER_ERROR('241'0)/*ERROR STATUS OVER-RUN*/
0308  DATA    USER_ERROR('242'0)/*RANDOM CHANNEL LIST NXM*/
0309  DATA    USER_ERROR('243'0)/*BUFFER OVER/UNDER -RUN*/
0310  DATA    USER_ERROR('244'0)/*BUFFER NXM*/
0311  DATA    USER_ERROR('245'0)/*USER STATUS WORD NXM*/
0312  DATA    USER_ERROR('246'0)/*USER STATUS WORD INV BUFR INDEX*/
0313  DATA    USER_ERROR('247'0)/*MASTER FIFO 7/8 FULL*/
0314  DATA    USER_ERROR('250'0)/*REQ TERM BY USER STAT WORD*/
0315  DATA    USER_ERROR('251'0)/*NO FIRMWARE FIFO BUFFER*/
0316  DATA    USER_ERROR('252'0)/*SLAVE FIFO 7/8 FULL*/
0317  DATA    USER_ERROR('253'0)/*RANDOM CHANNEL ADDRESS UNDER-RUN*/
0318  DATA    USER_ERROR('254'0)/*DATA UNDER-RUN*/
0319  DATA    USER_ERROR('255'0)/*CHANNEL/DEVICE NON-EXISTENT*/
0320  DATA    USER_ERROR('256'0)/*ILLEGAL ''USER ERROR'' CODE*/
0321  DATA    USER_ERROR('257'0)/*ILLEGAL ''USER ERROR'' CODE*/
0322  DATA    USER_ERROR('260'0)/*MULT EXT TRIG DIGITAL OUTPUT REQ*/
0323  DATA    USER_ERROR('261'0)/*ILLEGAL ''USER ERROR'' CODE*/
0324
0325
0326  CHARACTER*33  STATUS_RETURN(0:3)
0327  DATA    STATUS_RETURN(0)/*SUCCESSFUL INITIALIZED DATA XFER*/
0328  DATA    STATUS_RETURN(1)/*BUFFER FULL*/
0329  DATA    STATUS_RETURN(2)/*NON-FATAL BUFFER OVER-RUN*/

```

```
0330      DATA      STATUS_RETURN(3)/*ILLEGAL "STATUS RETURN" CODE
0331
0332
0333
0334
0335
0336
0337
0338
0339
0340      CALL FRCTOF (LUN)
0341      CALL DHEAD1 (LUN,'UBA LPA11K')
0342      CALL LINCHK (LUN,2)
0343      WRITE(LUN,10) CNTRL_MAINT
0344      FORMAT(' ',T8,'LA (S1',T24,Z8.4)
0345      10      CALL OUTPUT (LUN,CNTRL_MAINT,V1CNTRL_MAINT,0,0,0,'0')
0346      DO 20,I = 2,3
0347      FIELD = LIBSEXTZV(I,1,CNTRL_MAINT)
0348      IF (FIELD .NE. 0) THEN
0349
0350      CALL LINCHK (LUN,1)
0351
0352      WRITE(LUN,15) I + 14
0353      FORMAT(' ',T40,'EXTENDED BUS ADDRESS BIT ',I2,'.')
0354      ENDIF
0355
0356      20      CONTINUE
0357
0358      CALL OUTPUT (LUN,CNTRL_MAINT,V2CNTRL_MAINT,6,6,11,'0')
0359
0360      CALL OUTPUT (LUN,CNTRL_MAINT,V3CNTRL_MAINT,13,13,15,'0')
0361
0362      CALL LINCHK (LUN,1)
0363
0364      WRITE(LUN,25) CNTRL_STATUS
0365      FORMAT(' ',T8,'LA (S2',T24,Z8.4)
0366      25      FIELD = LIBSEXTZV(8,8,CNTRL_STATUS)
0367
0368      IF (FIELD .LT. '300'0) THEN
0369
0370      FIELD = LIBSEXTZV(0,3,CNTRL_STATUS)
0371
0372      CALL LINCHK (LUN,1)
0373
0374      WRITE(LUN,30) FIELD
0375      FORMAT(' ',T40,'USER INDEX = ',I1,'.')
0376      30      ENDIF
0377
0378      CALL OUTPUT (LUN,CNTRL_STATUS,V1CNTRL_STATUS,6,6,7,'0')
0379
0380      FIELD = LIBSEXTZV(8,8,CNTRL_STATUS)
0381
0382      IF (FIELD .NE. 0) THEN
0383
0384      CALL LINCHK (LUN,1)
0385
0386
```

```

0387
0388      IF (FIELD .GE. '340'0) THEN
0389
0390      40      WRITE(LUN,40) FATAL_ERROR(MIN('351'0, FIELD))
0391      FORMAT(' ',T40,A<<COMPRESSC (FATAL_ERROR(MIN('351'0, FIELD)))>>)
0392
0393      ELSE IF (FIELD .GE. '300'0) THEN
0394
0395      50      WRITE(LUN,50) START_ERROR(MIN('330'0, FIELD))
0396      FORMAT(' ',T40,A<<COMPRESSC (START_ERROR(MAX('330'0, FIELD)))>>)
0397
0398      ELSE IF (FIELD .GE. '240'0) THEN
0399
0400      60      WRITE(LUN,60) USER_ERROR(MIN('261'0, FIELD))
0401      FORMAT(' ',T40,A<<COMPRESSC (USER_ERROR(MIN('261'0, FIELD)))>>)
0402
0403      ELSE IF (FIELD .LE. '100'0) THEN
0404
0405      70      WRITE(LUN,70) STATUS_RETURN(MIN('3'0, FIELD))
0406      FORMAT(' ',T40,A<<COMPRESSC (STATUS_RETURN(MIN('3'0, FIELD)))>>)
0407      ENDIF
0408      ENDIF
0409
0410      CALL LINCHK (LUN,1)
0411
0412      90      WRITE(LUN,90) RDA
0413      FORMAT(' ',T8,'LA RDA',T24,Z8.4)
0414
0415      CALL LINCHK (LUN,1)
0416
0417      100     WRITE(LUN,100) (MAINTENANCE(I+J),J = 3,0,-1)
0418      FORMAT(' ',T8,'LA MR',T24,Z22.2)
0419
0420      IF (FIELD .EQ. '343'0) THEN
0421
0422      CALL LINCHK (LUN,2)
0423
0424      110     WRITE(LUN,110) MAINTENANCE(1)
0425      FORMAT(' ',T40,'EXPECTED STATUS = 052 (OCTAL)',/,1
0426      T40,'ACTUAL STATUS = ',03,' (OCTAL)')
0427
0428      ELSE IF (FIELD .EQ. '344'0) THEN
0429
0430      CALL LINCHK (LUN,2)
0431
0432      120     WRITE(LUN,120) (MAINTENANCE(I),I = 0,1)
0433      FORMAT(' ',T40,'MASTER VERSION NO. = ',13,'.',/,1
0434      T40,'SLAVE VERSION NO. = ',13,'.')
0435
0436      ELSE IF (FIELD .EQ. '346'0) THEN
0437
0438      IF (MAINTENANCE(0) .EQ. '200'0) THEN
0439
0440      CALL LINCHK (LUN,1)
0441
0442      125     WRITE(LUN,125)
0443      FORMAT(' ',T40,'SLAVE FIFO R/W ERROR')

```

0444
0445 ELSE IF (MAINTENANCE(1) .EQ. '201'0) THEN
0446
0447 CALL LINCHK (LUN,1)
0448
0449 130 WRITE(LUN,130)
0450 FORMAT(' ',T40,'KU11-K CLOCK A OVER-RUN')
0451 ENDIF
0452
0453 CALL LINCHK (LUN,1)
0454
0455 135 WRITE(LUN,135) MAINTENANCE(1)
0456 FORMAT(' ',T40,'SLAVE FIRMWARE SEQ = ',03,' (OCTAL)')
0457
0458 ELSE IF (FIELD .EQ. '347'0) THEN
0459
0460 CALL LINCHK (LUN,2)
0461
0462 140 WRITE(LUN,140) (MAINTENANCE(I),I = 0,1)
0463 FORMAT(' ',T40,'RETURNED VALUE = ',03,' (OCTAL)',/)
0464 1 T40,'EXPECTED VALUE = ',03,' (OCTAL)')
0465
0466 ELSE IF (FIELD .EQ. '350'0) THEN
0467
0468 CALL LINCHK (LUN,2)
0469
0470 150 WRITE(LUN,150) (MAINTENANCE(I),I = 0,1)
0471 FORMAT(' ',T40,'RETURNED VALUE = ',03,' (OCTAL)',/)
0472 1 T40,'EXPECTED VALUE = ',03,' (OCTAL)')
0473
0474 ELSE IF (FIELD .EQ. '322'0) THEN
0475
0476 CALL LINCHK (LUN,1)
0477
0478 155 WRITE(LUN,155) (MAINTENANCE(I),I = 1,0,-1)
0479 FORMAT(' ',T40,'ADDR OF MISSING DEV = ',06)
0480 ENDIF
0481
0482 if (emb\$w_hd_entry .ne. 98) then
0483
0484 if (ubareg(1) .ne. 0) then
0485
0486 call uba_datapath (lun,ubareg(1),ubareg(2))
0487 endif
0488 endif
0489
0490 call linchk (lun,1)
0491
0492 160 write(lun,160)
0493 format(' ',:)
0494
0495 call ucb\$1_char (lun,emb\$1_dv_char)
0496
0497 call ucb\$w_sts (lun,emb\$w_dv_sts)
0498
0499 call ucb\$1_opcnt (lun,emb\$1_dv_opcnt)
0500

```

0501      call ucb$w_errcnt (lun,emb$w_dv_errcnt)
0502      if (emb$w_hd_entry .ne. 98) then
0503          call linchk (lun,1)
0504          write(lun,160)
0505          call lpa11k_qio (lun,emb$w_dv_func)
0506          call irp$w_bcnt (lun,emb$w_dv_bcnt)
0507          call irp$w_boff (lun,emb$w_dv_boff)
0508          call irp$1_pid (lun,emb$1_dv_rapid)
0509          call irp$q_iosb (lun,emb$1_dv_iosb1)
0510      endif
0511
0512      RETURN
0513      END

```

PROGRAM SECTIONS

Name	Bytes	Attributes
0 \$CODE	1756	PIC CON REL LCL SHR EXE RD NOWRT LONG
1 \$PDATA	659	PIC CON REL LCL SHR NOEXE RD NOWRT LONG
2 \$LOCAL	2572	PIC CON REL LCL NOSHR NOEXE RD WRT LONG
3 EMB	512	PIC OVR REL GBL SHR NOEXE RD WRT LONG
Total Space Allocated	5499	

ENTRY POINTS

Address	Type	Name
0-00000000	LPA11K	

VARIABLES

Address	Type	Name	Address	Type	Name
3-00000052	I*4	CNTRL_MAINT	3-00000056	I*4	CNTRL_STATUS
3-00000010	L*1	EMBSB_DV_CLASS	3-00000010	L*1	EMBSB_DV_ERTCNT
3-00000011	L*1	EMBSB_DV_ERTMAX	3-0000003E	L*1	EMBSB_DV_NAMLNG
3-0000003A	L*1	EMBSB_DV_SLAVE	3-0000001D	L*1	EMBSB_DV_TYPE
3-00000036	I*4	EMBSL_DV_CHAR	3-00000012	I*4	EMBSL_DV_IOSB1
3-00000016	I*4	EMBSL_DV_IOSB2	3-00000026	I*4	EMBSL_DV_MEDIA
3-0000004E	I*4	EMBSL_DV_NUMREG	3-0000002E	I*4	EMBSL_DV_OPCNT
3-00000032	I*4	EMBSL_DV_OWNUIC	3-0000001E	I*4	EMBSL_DV_RQPID
3-00000000	I*4	EMBSL_HD_SID	3-0000003F	CHAR	EMBST_DV_NAME

3-00000024	I*2	EMBSW_DV_BCNT
-0000002C	I*2	EMBSW_DV_ERRCNT
-0000001A	I*2	EMBSW_DV_STS
-00000004	I*2	EMBSW_HD_ENTRY
-000007F0	I*4	FIELD
-000007F8	I*4	J
-0000005A	I*4	RDA

3-000000022	I*2	EMBSW_DV_BOFF
3-00000003C	I*2	EMBSW_DV_FUNC
3-00000002A	I*2	EMBSW_DV_UNIT
3-00000000E	I*2	EMBSW_HD_ERRSEQ
2-0000007F4	I*4	I
AP-00000004a	L*1	LUN

ARRAYS

Address	Type	Name	Bytes	Dimensions
3-00000000	L*1	EMB	512	(0:511)
3-00000052	I*4	EMBSL_DV_REGSAV	420	(0:104)
3-00000006	I*4	EMBSQ_HD_TIME	8	(2)
2-000000CA	CHAR	FATAL_ERROR	310	(224:233)
3-0000005E	L*1	MAINTENANCE	4	(0:3)
2-00000200	CHAR	START_ERROR	792	(192:215)
2-0000076A	CHAR	STATUS_RETURN	132	(0:3)
3-00000062	I*4	UBAREG	8	(2)
2-00000518	CHAR	USER_ERROR	594	(160:177)
2-00000000	CHAR	V1CNTRL_MAINT	7	(0:0)
2-000000A0	CHAR	V1CNTRL_STATUS	42	(6:7)
2-00000007	CHAR	V2CNTRL_MAINT	120	(6:11)
2-0000007F	CHAR	V3CNTRL_MAINT	33	(13:15)

LABELS

FUNCTIONS AND SUBROUTINES REFERENCED

Type	Name	Type	Name	Type	Name	Type	Name	Type	Name	Type	Name
I*4	COMPRESSC	I*4	DHEAD1		FRCTOF		IRPSL_PID		IRPSQ_IOSB		IRPSW_BCNT
IRPSW_BOFF	LIBSEXTZV	I*4	UCBSL_OPCNT		LINCHK		LPA11R_Q10		OUTPUT		UBA_DATAPATH
UCBSL_CHAR					UCBSW_ERRCNT		UCBSW_STS				

```
0001
0002
0003
0004 Subroutine LPA11K_QIO (lun,emb$w_dv_func)
0005
0006
0007 include 'src$:qicommon.for /nolist'
0271
0272
0273 byte lun
0274
0275 integer*2 emb$w_dv_func
0276
0277 integer*4 qicode(0:1,0:63)
0278
0279
0280
0281 if (qicode(0,0) .eq. 0) then
0282
0283 qicode(1,01) = %loc(ios_loadmcode)
0284
0285 qicode(1,02) = %loc(ios_startmproc)
0286
0287 qicode(1,04) = %loc(ios_initialize)
0288
0289 qicode(1,05) = %loc(ios_setclockp)
0290
0291 qicode(1,06) = %loc(ios_startdatap)
0292
0293 qicode(1,07) = %loc(ios_qstop)
0294
0295 qicode(1,55) = %loc(ios_setclock)
0296
0297 qicode(1,56) = %loc(ios_startdata)
0298
0299 do 10,i = 0,63
0300
0301 qicode(0,i) = 33
0302
0303 if (qicode(1,i) .eq. 0) then
0304
0305 qicode(1,i) = %loc(qio_string)
0306
0307 endif
10 continue
0309
0310
0311 call irp$w_func (lun,emb$w_dv_func,
0312 1 qicode(0,lib$extzv(0,6,emb$w_dv_func)))
0313
0314
0315
0316 end
```

PROGRAM SECTIONS

Name	Bytes	Attributes
0 \$'ODE	149	PIC CON REL LCL SHR EXE RD NOWRT LONG
1 \$PDATA	8	PIC CON REL LCL SHR NOEXE RD NOWRT LONG
2 \$LOCAL	548	PIC CON REL LCL NOSHR NOEXE RD WRT LONG
3 \$ICOMMON	1247	PIC OVR REL GBL SHR NOEXE RD WRT LONG
Total Space Allocated	1952	

ENTRY POINTS

Address	Type	Name
0-00000000	LPA11K_010	

VARIABLES

Address	Type	Name	Address	Type	Name
AP-00000008a	I*2	EMBSW DV FUNC	2-00000200	I*4	I
3-00000442	CHAR	IOS_ABORT	3-0000034D	CHAR	IOS_ACCESS
3-000003C2	CHAR	IOS_ACPCONTROL	3-000004B3	CHAR	IOS_AVAILABLE
3-00000297	CHAR	IOS_CLEAN	3-00000369	CHAR	IOS_CREATE
3-00000385	CHAR	IOS_DEACCESS	3-00000393	CHAR	IOS_DELETE
3-00000260	CHAR	IOS_DIAGNOSE	3-00000065	CHAR	IOS_DRVCLR
3-000004C9	CHAR	IOS_DSE	3-000000A9	CHAR	IOS_ERASETAPE
3-00000276	CHAR	IOS_FORMAT	3-00000071	CHAR	IOS_INITIALIZE
3-00000014	CHAR	IOS_LOADMCODE	3-000003A1	CHAR	IOS_MODIFY
3-000003E2	CHAR	IOS_MOUNT	3-00000000	CHAR	IOS_NOP
3-0000009D	CHAR	IOS_OFFSET	3-000000EB	CHAR	IOS_PACKACK
3-000000E0	CHAR	IOS_QSTOP	3-000003EF	CHAR	IOS_RDSTATS
3-00000421	CHAR	IOS_READCSR	3-00000169	CHAR	IOS_READHEAD
3-000002B6	CHAR	IOS_READLBLK	3-0000013F	CHAR	IOS_READPBLK
3-00000200	CHAR	IOS_READPRESET	3-00000195	CHAR	IOS_READTRACKD
3-0000033A	CHAR	IOS_READVBLK	3-0000045A	CHAR	IOS_READWTHBUF
3-00000484	CHAR	IOS_READWTHXBUF	3-0000004D	CHAR	IOS_RECAL
3-0000007C	CHAR	IOS_RELEASE	3-000001AB	CHAR	IOS_REREADN
3-000001B8	CHAR	IOS_REREADP	3-000000CA	CHAR	IOS_RETCENTER
3-000002E6	CHAR	IOS_REWIND	3-000002C9	CHAR	IOS_REWINDOFF
3-000000FC	CHAR	IOS_SEARCH	3-00000024	CHAR	IOS_SEEK
3-00000231	CHAR	IOS_SENSECHAR	3-00000309	CHAR	IOS_SENSEMODE
3-0000021D	CHAR	IOS_SETCHAR	3-000003B8	CHAR	IOS_SETCLOCK
3-00000088	CHAR	IOS_SETCLOCKP	3-000002DD	CHAR	IOS_SETMODE
3-000002ED	CHAR	IOS_SKIPFILE	3-000002FA	CHAR	IOS_SKIPRECORD
3-00000029	CHAR	IOS_SPACEFILE	3-0000010E	CHAR	IOS_SPACERECORD
3-000003D7	CHAR	IOS_STARTDATA	3-000000B4	CHAR	IOS_STARTDATAP
3-00000037	CHAR	IOS_STARTMPROC	3-0000020F	CHAR	IOS_STARTSPNDL
3-00000059	CHAR	IOS_STOP	3-0000000D	CHAR	IOS_UNLOAD
3-00000468	CHAR	IOS_WRITEBUFCRC	3-0000011E	CHAR	IOS_WRITECHECK
3-000001E4	CHAR	IOS_WRITECHECKH	3-000003FF	CHAR	IOS_WRITECSR
3-00000153	CHAR	IOS_WRITEHEAD	3-000002A2	CHAR	IOS_WRITELBLK
3-00000247	CHAR	IOS_WRITEMARK	3-00000314	CHAR	IOS_WRITEOF
3-0000012A	CHAR	IOS_WRITEPBLK	3-000001C9	CHAR	IOS_WITERET

LPA11K_QIO

M 7
16-Sep-1984 00:05:39
5-Sep-1984 14:00:17

VAX-11 FORTRAN V3.4-56

DISK\$VMSMASTER:[ERF.SRC]LPA11K.FOR;1

Page 11

3-0000017E CHAR IOS_WRITETRACKD
3-00000448 CHAR IOS_WRITEWTHBUF
AP-000000040 L*1 LUN

3-00000326 CHAR IOS_WRITEVBLK
3-00000257 CHAR IOS_WRTTMKR
3-000004A1 CHAR QIO_STRING

ARRAYS

Address	Type	Name	Bytes	Dimensions
2-00000000	I*4	QIOCODE	512	(0:1, 0:63)

LABELS

Address	Label
**	10

FUNCTIONS AND SUBROUTINES REFERENCED

Type	Name	Type	Name
	IRPSW_FUNC	I*4	LIB\$EXTZV

COMMAND QUALIFIERS

```
FORTRAN /LIS=LIS$:LPA11K/OBJ=OBJ$:LPA11K MSRC$:LPA11K  
/CHECK=(NOBOUNDS,OVERFLOW,NOUNDERFLOW)  
/DEBUG=(NOSYMBOLS,TRACEBACK)  
/STANDARD=(NOSYNTAX,NOSOURCE FORM)  
/SHOW=(NOPREPROCESSOR,NOINCLUDE,MAP)  
/F77 /NOG_FLOATING /I4 /OPTIMIZE /WARNINGS /NOD_LINES /NOCROSS_REFERENCE /NOMACHINE_CODE /CONTINJATIONS=19
```

COMPILATION STATISTICS

Run Time:	7.64 seconds
Elapsed Time:	17.17 seconds
Page Faults:	280
Dynamic Memory:	209 pages

0150 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY